



HAL
open science

Image Processing on Mobile Devices: An Overview

Rafika Thabet, Ramzi Mahmoudi, Mohamed Hedi Bedoui

► **To cite this version:**

Rafika Thabet, Ramzi Mahmoudi, Mohamed Hedi Bedoui. Image Processing on Mobile Devices: An Overview. 2014 First International Image Processing, Applications and Systems Conference (IPAS), Nov 2014, Sfax, Tunisia. 10.1109/IPAS.2014.7043267 . hal-03112423

HAL Id: hal-03112423

<https://hal.science/hal-03112423v1>

Submitted on 16 Jan 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Image Processing on Mobile Devices: An Overview

Rafika THABET¹, Ramzi MAHMOUDI^{1,2} and Mohamed Hedi BEDOUI¹

¹Laboratoire Technologie Imagerie Médicale - LTIM-LR12ES06
Faculté de Médecine de Monastir - 5019 Monastir, Tunisie

²IGM, Unité Mixte CNRS-UMLV-ESIEE UMR8049, University Paris-Est
Cité Descartes, BP99, 93162 Noisy Le Grand, France

rafika.thabet@gmail.com, ramzi.mahmoudi@esiee.fr, medhedi.bedoui@fmm.rnu.tn

Abstract—Image processing technology has grown significantly over the past decade. Its application on low-power mobile devices has been the interest of a wide research group related to newly emerging contexts such as augmented reality, visual search, object recognition, and so on. With the emergence of general-purpose computing on embedded GPUs and their programming models like OpenGL ES 2.0 and OpenCL, mobile processors are gaining a more parallel computing capability. Thereby, the adaptation of these advancements for accelerating mobile image processing algorithms has become actually an important topical issue. In this paper, our interest is based on reviewing recent challenging tasks related to mobile image processing using both serial and parallel computing approaches in several emerging application contexts.

Keywords—image processing, mobile devices; mobile GPUs; multi-threading; mobile GPGPU; OpenGL ES 2.0; OpenCL

I. INTRODUCTION

Image processing allows extracting the meaning of an observed scene from the acquired information. Nowadays, there is a great interest in image processing algorithms able to work on mobile platforms [1]. Indeed, incorporating image processing capabilities on mobile devices open new opportunities in different application contexts such as augmented reality, visual search, object recognition, and so on. However, the high computational complexity of some image processing algorithms and both their long processing time and energy consumption prevent them from being effectively used in real-time mobile applications. Thus, the practice of these algorithms on such a device is still a challenging task since these devices are typically limited by: power supply, battery capacity, energy consumption, computational power, RAM amount, etc.

In the past few years, exploring the use of desktop GPUs as a general-purpose co-processor [2] to accelerate compute-intensive applications has been an active research subject. Several speedups have been reported in the literature, depending on the applications, the algorithms parallelism, and the computing capability provided by the GPUs. Recently, these programmable GPUs have proved their feasibility on mobile devices [1], such as smartphones and tablets. In fact, rapid advancements of mobile computational capabilities and memory specifications have allowed making processor-intensive applications more possible, which were considered infeasible just a few years ago. So, as the GPUs have become

an integrated component with a multi-core architecture in mobile devices, researchers have explored the opportunities of using the low-power mobile GPUs as a general-purpose accelerator, similar to its role in a desktop [3], by diverting their limits. Therefore, General-Purpose computing on GPUs (GPGPU) for mobile devices has become possible [1], which has opened new opportunities to speed up mobile image processing algorithms. In addition, several emerging programming models – such as Open Graphics Library for Embedded System 2.0 (OpenGL ES 2.0) [4] and Open Computing Language (OpenCL) [5] – for the mobile GPGPU computing have become recently supported by various mobile processors.

Eventually, our interest is to survey recent researches related to image processing running on low-power mobile devices in order to have a new idea about the most and least extensively studied application areas, algorithms and concepts explored for each area, improvements made by exploiting parallel computing, and speedups achieved using programming models for mobile GPGPU. Accordingly, researchers will be guided in their future choices.

II. STATE OF THE ART

A. Mobile-serial-computing approaches

Image processing on mobile platforms has become an area of research that keeps an important key to future advances in augmented reality, visual search, object recognition, and several other application domains. This section will be devoted to present the recent works of the previously mentioned fields using mainly serial computation on mobile devices.

1) Mobile Augmented Reality

Augmented Reality (AR) is a type of virtual reality that allows seeing the real world and the virtual objects together by superimposing virtual objects upon the real world. So, the Mobile Augmented Reality (MAR) system is designed specifically for mobile platforms that must track objects, recognized from a database. A 2009 overview on the tracking history for the MAR was presented in [6]. Indeed, the MAR algorithms were divided into two types: the first one used natural features and the second one used artificial markers.

For the first type, several MAR systems based on visual feature recognition and tracking [7, 8, 9, 10, 11, 12, 13] have been recently proposed. Most of these research efforts have

focused on improving the tracking speed for the MAR. In fact, in [7] the speed was improved by tracking Speeded-Up Robust Features (SURF) [14] in constrained locations. In addition, in [8, 9, 10, 11, 12] significant advancement were made in pose tracking to respond to strict real time constraints. Actually, in [8] two techniques for natural feature tracking from planar targets were presented in a real-time way. These techniques used an approach based on the modified descriptors: Scale Invariant Feature Transform (SIFT) [15] and Ferns [16]. This work presented the first fully self-contained natural feature tracking system able to track the full six degrees of freedom at up to 20 Hz of real-time frame rates from natural features using only the integrated camera phone. The latter work was first resumed in [9] then in [10], where a template-matching-based tracker was used at frame rates of up to 30 Hz on mobile phones of their generation. The resumed approaches increased the performance and the robustness of the initial approach.

Yet, in [11] a method for real-time creation and tracking of panoramic maps was presented. This method ran also on mobile phones at 30Hz and the generated maps allowed a drift-free rotation tracking in outdoor scenarios. Whereas, in [12] a real-time parallel tracking and mapping was presented with a monocular camera in a small workspace. That work was an adaptation of a simultaneous localization and mapping approach on mobile devices.

However, the scalability of the MAR systems was not addressed by any of these last efforts. By contrast, in [13] a binary descriptor, called Local Difference Binary (LDB), was introduced to facilitate the MAR scalability. This descriptor was computed by using the integral image technique. For a large database, the LDB achieved a greater accuracy and a faster matching speed than the Binary Robust Independent Elementary Features (BRIEF) [17] descriptor.

For the second type, few marker-based MAR systems [18, 19] were suggested in the literature. In [18] a successor of the popular ARToolkit marker tracking library [20] was presented for the use of the AR on mobile devices such as smartphones, PDAs and Ultra Mobile PCs. It was also called ARToolkitPlus. Besides, in [19] a real-time AR program was implemented on a smart phone by using the OpenCV library [16], the ARToolkitPlus and the VRTToolkit [21].

2) Mobile Visual Search

A Mobile Visual Search (MVS) is a type of search engine designed specifically for mobile devices. In Mobile Image Search (MIS), through a query image taken with the mobile phone or using certain keywords, any information can be found on the Internet. Therefore, the content, shape, texture and color of the image are used to be compared to a database, and then the approximate results from the query are delivered.

Various recent MVS systems have been developed to search for camera phone images of CD covers [22, 23, 24], photos [25], printed documents [26, 27, 28], locations [29, 30, 31], and so on [32, 33]. Most of these studies were interested in performing a visual search on mobile devices.

In the aim of searching for mobile images of CD covers, in [22] a mobile image searching method was put forward by consistently exploiting visual and spatial information to

improve feature-discriminative power. Also, in [23] a CD cover search system was used to compare different local descriptors. As a result, the SIFT descriptor was broadly accepted as the best performed feature. However, in a low-bit transmission, a Compressed Histogram of Gradients (CHoG) [34] descriptor had an advantage over the SIFT. Nevertheless, in [24], an image segmentation technique was applied to a CD cover retrieval system on mobile devices. Given a mobile-device image, this technique used edge detection and region merging mechanisms to extract the interest region from a complex background scene. The suggested method consisted in: initially, an image segmentation by a mean-shift [35] algorithm; next, an automatic region merging; and then, an object-contour extraction by the labeled regions as either foreground or background. Compared with other automatic segmentation methods and without the user interaction, the proposed method demonstrated its efficiency.

However, in [25] a complete photo-to-search overview was given, starting with the architecture of an efficient mobile system until arriving at the framework of an image recognition algorithm.

In the goal of searching for mobile images of printed documents [26], a hybrid book recognition system on a bookshelf was suggested for use in book management systems. This approach consisted of the spine recognition pipeline based on combining both text and image features of the book spine image. In the case of text-based recognition system, a high recall at low precision was achieved, and in the case of an image feature-based recognition system, a moderate recall at high precision was attained. In the same context, in [27] a mobile printed-document retrieval system was presented using both text and image-based features. An algorithm based on edge-enhanced Maximally Stable Extremal Regions (MSER) [36] was used for text detection, a gradient-based algorithm was employed for rectifying the extracted title text image patch, and an optical character recognition was used for recognition during image-feature extraction. Both text and image-based features were transmitted to a server. The title text was used to make an online search and the image-based features were used to check the search results. The proposed system was able to perform a web-scale document search using the title text and to achieve an accuracy of retrieved documents using the image-based features. The latter work was resumed in [28], where the system was improved by using a geometric verification framework incorporating both text and image-feature information.

In order to search landmarks and locations, the authors in [29] put forward a discovery of disappearance points and an identification of facades; whereas, the authors in [30] used a fusion of multiple image representations, with an aligned facade and viewpoint, to improve the search accuracy. However, the authors in [31] presented an active query-sensing system to help users to take the best second query for an ulterior location search, which was not addressed by any previous work. Actually, it was about an interactive mobile vision system.

Similarly, in [32] a multimodal interactive image search system, called Joint search with ImaGe, Speech And Word

(JIGSAW), was described on mobile devices. This system took advantage of the multimodal input and multi-touch natural user interactions of mobile devices. The JIGSAW was deployed and evaluated on a real-world phone system. After that, this system was resumed in [33] to achieve a better gain in terms of time and search performance. The proposed algorithm was therefore called JIGSAW+ for Joint search with Image, Speech And Word Plus.

3) Mobile Object Recognition

Mobile Object Recognition (MOR) is a fundamental task for many mobile computer vision applications, such as MAR, MIS, and so on. For realizing this case study, several researches exist in literature which allow recognizing food ingredients [37], characters [38], faces [39, 40, 41, 42, 43, 44, 45, 46], objects [47, 48, 49, 50], etc. The focus investigation of all these studies was to prove the feasibility of performing object recognition systems on mobile platforms.

Few researchers have been interested in recognizing food ingredients. Accordingly, in [37] a system of recipe recommendation on mobile devices was carried out allowing extracting color feature, recognizing 30 kinds of food ingredients, and recommending food recipes.

Also, few researches have focused on recognizing characters from mobile-platform images. In order to develop a business card reader, in [38] a character segmentation technique was presented for business card images taken using a mobile phone camera. The proposed technique consists in extracting text regions from card images and segmenting them into characters, which was successfully implemented on a moderately powerful notebook.

The practice of face detection and recognition on mobile platforms have become increasingly frequent in the literature [39, 40, 41, 42, 43, 44, 45, 46], where most of them use the OpenCV library. Thus, in [39] a cascade-subspace face/eye detector was put forward and compared to an AdaBoost [51] detector whose results showed that at a comparable speed the suggested detector could detect eyes at more precise locations. However, in [40, 41, 42] face-recognition systems using the OpenCV library were proposed. These systems are based on the AdaBoost algorithm for the face detection stage. A real-time Eigenfaces algorithm [52] and a Local Binary Pattern (LBP) [53] were used, respectively in [40] and [41, 42], for the face recognition stage. Nevertheless, in [43] the face recognition performance was improved on a mobile phone by using a selective method that generated Gabor features [54] based on a contribution measurement got by a discriminant analysis. Likewise, in [44] the skin-color image segmentation was presented as a preprocessing stage for human-face detection on a mobile platform. The RGB-H-CbCr skin-color model introduced in [55] was applied with few minor adjustments for a better result. Indeed, compared with few other types of skin-color models, the suggested model showed better results of skin extraction. Nonetheless, in [45, 46] web services were used for improving the face recognition system on Android platforms. So, in [45], besides the face recognition process using a locally-implemented-Eigenfaces decomposition, a web service using SOAP messages [56] was also developed for a more advanced feature extraction and

face-image classification. On the other hand, in [46] a cloud computing service [57] with a Representation State Transfer (REST) [58] communication and an Android face detector API, as a library, were used for performing the recognition system.

To recognize objects, in [47] local-feature descriptors and their matching method were devised for recognizing registered objects in a real-time way. Yet, in [48] an accelerated MOR system was presented using an adapted version of the SURF which was based on two new techniques: content-aware tiling and gradient-moment-based orientation assignment. The first technique improved the data locality and reduced the memory traffic, whereas the second avoided the penalties caused by pipeline hazards. Thus, the performance and robustness of the suggested techniques were evaluated on three mobile platforms and compared with the original SURF algorithm, where the accelerated SURF achieved a higher speedup without affecting the recognition accuracy.

In the context of interactive image segmentation for mobile object recognition, such researches have been proposed [49, 50]. Also, the authors in [49] performed the tracking of multiple seeds that improved the segmentation results. Those seeds were points indicated by the real-world user, following a laser pointer on a smartphone. The implementation of this method, which was based on the optical flow algorithm of Lucas-Kanade [59] and the Fast Multi Object Fuzzy Segmentation (Fast-MOFS) [60] algorithm, proved its real-time feasibility with high frame rates on limited-resource devices. However, in [50] the proposed algorithm starts with pre-segmentation by the mean-shift algorithm, followed by merging regions using discriminative clustering and completed by a local-neighborhood-region classification and pruning. Compared to the method GraphCut [61], with pre-segmented regions using the Watershed algorithm [62], and to the Maximal-Similarity-based Region Merging (MSRM) [63], the proposed method had better quality results.

On the other hand, in order to compare the performances of the following mobile platforms: Nokia N900, LG Optimus One and Samsung Galaxy SII, the authors in [64] ported and tested few classic computer vision algorithms by using the OpenCV Library. These algorithms were dedicated to three tasks: feature extraction, face detection, and image segmentation. For the first task, three algorithms were employed: Features from Accelerated Segment Test (FAST) [65], a Modified Upright SURF (MU-SURF) [66], and the SURF. Concerning the second task, the AdaBoost algorithm was used. For the final task, the GraphCut algorithm was considered.

B. Mobile-parallel-computing approaches

In this section, we investigate the computational capability and energy efficiency of mobile CPU-GPU systems to accelerate image processing applications. In this connection, we give an existing and recent overview of mobile parallel computing utilizing both multi-threading and GPGPU concepts.

1) Multi-threading concept on mobile devices

In order to achieve optimizations, many developers have resorted to the use of the multi-threading concept in the MAR

[67, 68, 69], Mobile Food Recognition (MFR) [70, 71, 72], Mobile Optical Character Recognition (MOCR) [73] contexts.

Some researchers [67, 68, 69] have been interested in optimizing the MAR context by using a multi-threading model. In [67] the workloads of the MAR system, running on a low-power Intel® Atom™ processor of Mobile Internet Devices (MIDs), were analyzed. After identifying the hotspot functions that took the largest part of the overall response time, an architectural characterization of those hotspot functions was presented in terms of Cycles Per Instruction (CPI), Misses Per Instruction (MPI), etc. Also, implementing several software optimizations – such as multi-threading, vectorization, cache conflict avoidance, and various code optimizations that reduce the number of computations – improved 3X the performance in the overall execution time. The last work was resumed in [68] to show the benefits of hardware accelerations for the MAR that needed important computation processing for object recognition and matching. Therefore, two proposed hardware accelerations were designed: one for object recognition and the other for match processing. As a result, those hardware accelerations significantly enhanced the overall response time by 7X. Similarly, in [69] a MAR system of tourist guides was presented. This system combined the camera, location, orientation and motion sensors to put forward a new manner to bring more tourism information to users. So, to track objects of interest in the live camera view and to place augmented information on top of them, a combination of motion estimation algorithms and orientation sensors was used. Significant speedups were attained by identifying the hotspots in the MAR source codes and using the following optimizations: resolving hotspots by multi-threading approach, converting data and computation type, and vectorization.

In the context of the MFR, enough research efforts have been presented in [70, 71, 72]. All these works are focused on improving the speed of their MFR systems by using the multi-threading concept. All of them had the same purposes, which are: (i) implementing an interactive and real time MFR system running on the Android smartphone Samsung Galaxy Note II (1.6GHz Quad Core), (ii) estimating calories and nutritious food, (iii) recording the user's eating habits, (iv) adopting a linear SVM and an histogram χ^2 kernel based on kernel-feature maps [74], (v) and implementing a multi-threaded system on Quad CPU cores. Furthermore, in these works, an automatic adjustment was applied by using the GrabCut [75] method. After that, the user drew a bounding box over the food region. However, the authors in [70] used a color histogram and the SURF-based bag of features for the food recognition stage; whereas, the authors in [71, 72] adopted a Fisher Vector (FV) [76], an Histogram of Oriented Gradients (HOG) [77] patches, and color patches. According to the experiments of recognition accuracy and processing time, the proposed method in [70] has been the least efficient of all. Note that [71] and [72] offered the same method but the second one gave a more detailed study with several versions and more tests of the suggested method.

On the other hand, in [73] a reference implementation of the MOCR workload was analyzed, on the low-power Intel® Atom™ processor for handheld devices. Also, the elementary

hotspot functions that incurred most of the overall response time were identified. A detailed architectural characterization of those hotspot functions, in terms of CPI, MPI, and various phases of memory bandwidth, were also presented. Moreover, several software/algorithmic optimizations – such as multi-threading, image sampling for a hotspot function, and various code optimization – as well as hardware acceleration were implemented and analyzed. The optimization results could significantly improve the overall processing time and the power consumption.

2) GPGPU concept on mobile devices

The emerging and advent of the GPUs with programmable shaders on mobile devices like smart phones and tablets have motivated developers to utilize the GPUs to offload computationally intensive tasks and relieve the burden from the mobile CPU. In fact, the capability of the GPGPU on mobile devices has opened a new era for mobile computing and has enabled many computationally demanding mobile image processing algorithms [1]. Recently, programming models supporting mobile GPGPU computing such as OpenGL ES 2.0 and OpenCL, has become available on most mobile devices. Thus, in this section, we review the recent studies that use each of these frameworks for controlling mobile-GPU tasks.

a) Using the OpenGL ES 2.0 programming model

Originally, the OpenGL ES programming model was designed for 3D-graphics rendering for embedded systems. From version 2.0, the OpenGL ES framework has become able to support programmable shaders. Thus, most of recent researches have benefited from this version so as to harness the computation power of the mobile battery-powered GPUs. Their special investigation has focused on the role of mobile GPUs for energy or/and time optimization in real-time applications. Accordingly, we present recent studies using the OpenGL ES 2.0 shading language to support the GPGPU on mobile platforms for optimizing image processing algorithms.

In the literature, few research efforts [78] are interested in accelerating edge-detection algorithms. Nevertheless, most of them [79, 80, 81, 82, 83, 84, 85, 86] have focused on accelerating corner-detection algorithms.

For the edge detection context, in [78] a programmable shader implementation of a Canny edge detection [87] was presented for measuring the performance of the GPUs on a range of devices. The Open Graphics Library Shading Language (GLSL) [4] from the OpenGL ES 2.0 was used. The purpose was to determine the advantages of utilizing mobile GPUs for image analysis instead of the usual processing performed entirely on the mobile CPU.

However, much effort [79, 80, 81, 82, 83, 84, 85, 86] has focused on corner detection. Indeed, in [79] the FAST corner detection algorithm was implemented using parallel computing on GPUs with the GLSL tool. The results increased significantly the computational speed. Even compared to the SURF algorithm, the speed of the proposed method is much faster. In addition, the suggested method was well performed

for processing different images and using various mobile devices.

Yet, in [80] a GPGPU implementation of a modified SURF descriptor (called uSURF-ES) was presented. The proposed method used a shader generator to adjust the variations in the GPU capabilities. The main contribution was to prove the feasibility of modern mobile-graphic accelerators for GPGPU tasks, particularly for the detection phase in the natural-feature tracking used in the MAR applications. So to determine the speedup, the mobile-GPU implementation of the uSURF-ES was compared to the mobile-CPU implementation of the SURF in the OpenCV library.

In [81] the main elements, underlying the implementation, evaluation and optimization of computer-vision and image-processing algorithms, were analyzed. First, the characteristics of the embedded GPUs and their limits compared to the embedded CPU, were presented. Secondly, techniques with an optimized shader design were put forward to achieve enhanced performances. In order to prove the validity and effectiveness of the proposed techniques, three algorithms – which are: cartoon-style Non-Photorealistic Rendering (NPR) [88], Belief propagation (BP) stereo matching [89], and SURF – were selected and implemented on embedded GPUs. Finally, the performance between the implementation on embedded CPU and GPUs was evaluated in terms of the achieved speedup and execution time. Note that an initial version of the latter work had been presented in [82] where a Harris corner detection [90] algorithm was implemented instead of the SURF algorithm. In the same context, in [83] a set of metrics to measure the characteristics were presented for image processing algorithms running on mobile phone GPUs. These measurements allow assisting users in the design and implementation stage as well as in the bottlenecks classification. Also, techniques with an optimized shader design were proposed to obtain an increased performance. As a case study, the algorithms used in [81] were also employed to prove the effectiveness of the suggested techniques.

In [84] an implementation of the SIFT algorithm was presented using the GPU acceleration in mobile devices. The main contributions resided in: first, profiling the major stages of the algorithm on both mobile CPU and GPUs which allowed the development of a dataflow schema describing the methodical partitioning of the workload between the mobile CPU and GPUs; and second, reordering and compressing the input image to minimize the communication overhead of the CPU-GPUs memory transfers and to accelerate the GPU computation. The performances of the proposed implementation were tested using various mobile devices. A considerable speedup was achieved unlike an optimized CPU version running in a single thread and unlike a GPU implementation published in [91]. In addition, total energy and power consumption were greatly reduced. A significant speedup was achieved over an optimized CPU version and related GPU work, resulting in a near real-time detection.

Moreover, few works [85, 86] has been interested in computing power and energy consumption running on a mobile CPU-GPUs platform for face recognition, as a case study. Indeed, in [85] the first accelerated implementation of

the LBP feature extraction for a face tracking approach was presented. The two contributions of the latter research are: describe the challenges of designing on mobile GPUs, and prove the performance achieved for mobile image recognition applications on mobile GPUs in terms of speedup and power consumption in comparison to using the mobile CPU on the same platform. However, in [86] a comparison between the performance and power efficiency of mobile GPUs, a mobile CPU, and a desktop GPU was made. An accelerated Gabor face feature extraction on mobile GPUs was implemented as a case study, and the results confirmed that utilizing the mobile GPUs could achieve a significant performance speedup and a substantial energy reduction, but a slight power increase, for the proposed application.

b) Using the OpenCL programming model

The OpenCL is a new standard of the GPGPU tool for mobile GPUs. Unlike the OpenGL ES 2.0, the OpenCL is a pure parallel computing library on heterogeneous platforms including the CPU, the GPU, and even the DSP.

In the literature, few studies [92, 93, 94] have been interested in implementing general-purpose computing using an OpenCL framework on mobile GPUs. Indeed, in [92] an OpenCL Embedded Profile prototype emulated by the OpenGL ES was demonstrated and its advantages in performance and energy efficiency were also shown. However, to the best of our knowledge, the work presented in [93] introduced the first study of the GPGPU computing capability using the OpenCL framework on real mobile GPUs. Therefore, the indicated work suggested an accelerated object removal algorithm, as a case study. The results showed that offloading the core computations to mobile GPUs using the OpenCL framework could significantly reduce the processing time, hence the feasibility of intensive computing algorithms such as computer vision for mobile devices. Recently, the latter work has been resumed in [94] with more tests and details.

Newly, in [95] a comparison between the GLSL and the OpenCL as GPGPU software platforms has been presented in terms of implementation efficiency and speed performance. Another comparison between the mobile GPUs and CPU can also be found in the same reference. As a case study, the feature detection algorithms, SIFT and SURF, have been parallelized and optimized on mobile GPUs. Accordingly, the results have shown that the mobile GPUs has significantly surpassed the comparable mobile CPU on the same platforms. The implementation using the GLSL has been less efficient even if it has a comparable performance with the OpenCL. Furthermore, the power consumption on each implementation has been measured and compared, which has shown that the GLSL has consumed more energy than the OpenCL.

III. CONCLUSION

The ability of mobile platforms for image processing seems to have achieved a tipping point since 2010, mainly because several devices could benefit from offloading the processing to the GPUs.

Despite the fact that some image processing algorithms expressing a high degree of parallelism have been successfully accelerated, using the OpenGL ES 2.0 and the OpenCL frameworks, the emerging applications need more speedups. So, efficient implementations of image processing techniques are still remaining challenging tasks.

In order to efficiently and fully use the limited computation resources on mobile processors, some points should be carefully considered to achieve a high performance: (i) explore the algorithmic parallelism, (ii) partition the tasks between the CPU and the GPUs, (iii) optimize the overhead of CPU-GPUs memory transfers, (iv) avoid useless access to the memory (v) choose the appropriate programming model for using the mobile GPGPU, (vi) and select the right hardware design including a unified shader design, a tiling architecture, and a texture compression.

According to this recent overview, to the best of our knowledge, none of the mentioned researches have taken into consideration the set of these points; so it will be interesting to impose selection criteria in order to discover an efficient parallelization strategy of image processing algorithms running on low-power mobile GPUs, englobing all these points.

IV. REFERENCES

- [1] K. Pulli, A. Baksheev, K. Korniyakov, and V. Eruhimov, "Real-time computer vision with opencv," *Communications of the ACM*, vol. 55, no. 6, pp. 61–69, 2012.
- [2] J. D. Owens, M. Houston, D. Luebke, S. Green, J. E. Stone, and J. C. Phillips, "Gpu computing," *Proceedings of the IEEE*, vol. 96, no. 5, pp. 879–899, 2008.
- [3] T. Akenine-Moller and J. Strom, "Graphics processing units for handhelds," *Proceedings of the IEEE*, vol. 96, no. 5, pp. 779–789, 2008.
- [4] A. Munshi and J. Leech, "the khronos group, the opengl es specification," [Online]. Available: <http://www.khronos.org/opengles>
- [5] A. Munshi, "the khronos group, the opencl specification," [Online]. Available: <http://www.khronos.org/opencl>
- [6] D. Wagner and D. Schmalstieg, "History and future of tracking for mobile phone augmented reality," in *International Symposium on Ubiquitous Virtual Reality*, pp. 7–10, Citeseer, 2009.
- [7] D.-N. Ta, W.-C. Chen, N. Gelfand, and K. Pulli, "Surfrac: Efficient tracking and continuous object recognition using local feature descriptors," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pp. 2937–2944, IEEE, 2009.
- [8] D. Wagner, G. Reitmayr, A. Mulloni, T. Drummond, and D. Schmalstieg, "Pose tracking from natural features on mobile phones," in *Proceedings of the 7th IEEE/ACM International Symposium on Mixed and Augmented Reality*, pp. 125–134, IEEE Computer Society, 2008.
- [9] D. Wagner, D. Schmalstieg, and H. Bischof, "Multiple target detection and tracking with guaranteed framerates on mobile phones," in *Mixed and Augmented Reality, 2009. ISMAR 2009. 8th IEEE International Symposium on*, pp. 57–64, IEEE, 2009.
- [10] D. Wagner, G. Reitmayr, A. Mulloni, T. Drummond, and D. Schmalstieg, "Real-time detection and tracking for augmented reality on mobile phones," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 16, no. 3, pp. 355–368, 2010.
- [11] D. Wagner, A. Mulloni, T. Langlotz, and D. Schmalstieg, "Real-time panoramic mapping and tracking on mobile phones," in *Virtual Reality Conference (VR), 2010 IEEE*, pp. 211–218, IEEE, 2010.
- [12] G. Klein and D. Murray, "Parallel tracking and mapping on a camera phone," in *Mixed and Augmented Reality, 2009. ISMAR 2009. 8th IEEE International Symposium on*, pp. 83–86, IEEE, 2009.
- [13] X. Yang and K.-T. Cheng, "Ldb: An ultra-fast feature for scalable augmented reality on mobile devices," in *Mixed and Augmented Reality (ISMAR), 2012 IEEE International Symposium on*, pp. 49–57, IEEE, 2012.
- [14] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in *Computer Vision—ECCV 2006*, pp. 404–417, Springer, 2006.
- [15] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.
- [16] G. Bradski and A. Kaehler, *Learning OpenCV: Computer vision with the OpenCV library*. O'Reilly Media, Inc., 2008.
- [17] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "Brief: Binary robust independent elementary features," in *Computer Vision—ECCV 2010*, pp. 778–792, Springer, 2010.
- [18] D. Wagner and D. Schmalstieg, "Artoolkitplus for pose tracking on mobile devices," in *Proceedings of 12th Computer Vision Winter Workshop (CVWW'07)*, pp. 139–146, 2007.
- [19] J. Kim and H. Jun, "Implementation of image processing and augmented reality programs for smart mobile device," in *Strategic Technology (IFOST), 2011 6th International Forum on*, vol. 2, pp. 1070–1073, IEEE, 2011.
- [20] H. Kato and M. Billinghurst, "Marker tracking and hmd calibration for a video-based augmented reality conferencing system," in *Augmented Reality, 1999. (IWAR'99) Proceedings. 2nd IEEE and ACM International Workshop on*, pp. 85–94, IEEE, 1999.
- [21] B. Loulier, "Virtual reality on iphone," [Online]. Available: <http://www.benjaminloulier.com/posts/virtual-reality-on-iphone-code-inside>
- [22] X. Liu, Y. Lou, A. W. Yu, and B. Lang, "Search by mobile image based on visual and spatial consistency," in *Multimedia and Expo (ICME), 2011 IEEE International Conference on*, pp. 1–6, IEEE, 2011.
- [23] V. Chandrasekhar, D. M. Chen, A. Lin, G. Takacs, S. S. Tsai, N.-M. Cheung, Y. Reznik, R. Grzeszczuk, and B. Girod, "Comparison of local feature descriptors for mobile visual search," in *Image Processing (ICIP), 2010 17th IEEE International Conference on*, pp. 3885–3888, IEEE, 2010.
- [24] Y. Liu, H. Zhang, L. Chai, and Y. Qi, "A foreground segmentation method for mobile image retrieval system," in *Network Infrastructure and Digital Content (IC-NIDC), 2012 3rd IEEE International Conference on*, pp. 492–497, IEEE, 2012.
- [25] B. Girod, V. Chandrasekhar, D. M. Chen, N.-M. Cheung, R. Grzeszczuk, Y. Reznik, G. Takacs, S. S. Tsai, and R. Vedantham, "Mobile visual search," *Signal Processing Magazine, IEEE*, vol. 28, no. 4, pp. 61–76, 2011.
- [26] S. S. Tsai, D. Chen, H. Chen, C.-H. Hsu, K.-H. Kim, J. P. Singh, and B. Girod, "Combining image and text features: a hybrid approach to mobile book spine recognition," in *Proceedings of the 19th ACM international conference on Multimedia*, pp. 1029–1032, ACM, 2011.
- [27] S. S. Tsai, H. Chen, D. Chen, G. Schroth, R. Grzeszczuk, and B. Girod, "Mobile visual search on printed documents using text and low bit-rate features," in *Image Processing (ICIP), 2011 18th IEEE International Conference on*, pp. 2601–2604, IEEE, 2011.
- [28] S. S. Tsai, H. Chen, D. Chen, R. Vedantham, R. Grzeszczuk, and B. Girod, "Mobile visual search using image and text features," in *Signals, Systems and Computers (ASILOMAR), 2011 Conference Record of the Forty Fifth Asilomar Conference on*, pp. 845–849, IEEE, 2011.
- [29] G. Baatz, K. Köser, D. Chen, R. Grzeszczuk, and M. Pollefeys, "Handling urban location recognition as a 2d homothetic problem," in *Computer Vision—ECCV 2010*, pp. 266–279, Springer, 2010.
- [30] D. M. Chen, G. Baatz, K. Koser, S. S. Tsai, R. Vedantham, T. Pylvanainen, K. Roimela, X. Chen, J. Bach, M. Pollefeys, et al., "City-scale landmark identification on mobile devices," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pp. 737–744, IEEE, 2011.
- [31] F. X. Yu, R. Ji, and S.-F. Chang, "Active query sensing for mobile location search," in *Proceedings of the 19th ACM international conference on Multimedia*, pp. 3–12, ACM, 2011.

- [32] H. Li, Y. Wang, T. Mei, J. Wang, and S. Li, "Interactive multimodal visual search on mobile device," *Multimedia, IEEE Transactions on*, vol. 15, no. 3, pp. 594–607, 2013.
- [33] Y. Wang, T. Mei, J. Wang, H. Li, and S. Li, "Jigsaw: interactive mobile visual search with multimodal queries," in *Proceedings of the 19th ACM international conference on Multimedia*, pp. 73–82, ACM, 2011.
- [34] V. Chandrasekhar, G. Takacs, D. Chen, S. Tsai, R. Grzeszczuk, and B. Girod, "Chog: Compressed histogram of gradients a low bit-rate feature descriptor," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pp. 2504–2511, IEEE, 2009.
- [35] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 24, no. 5, pp. 603–619, 2002.
- [36] H. Chen, S. S. Tsai, G. Schroth, D. M. Chen, R. Grzeszczuk, and B. Girod, "Robust text detection in natural images with edge-enhanced maximally stable extremal regions," in *Image Processing (ICIP), 2011 18th IEEE International Conference on*, pp. 2609–2612, IEEE, 2011.
- [37] T. Maruyama, Y. Kawano, and K. Yanai, "Real-time mobile recipe recommendation system using food ingredient recognition," in *Proceedings of the 2nd ACM international workshop on Interactive multimedia on mobile and portable devices*, pp. 27–34, ACM, 2012.
- [38] A. F. Mollah, S. Basu, and M. Nasipuri, "Segmentation of camera captured business card images for mobile devices," *arXiv preprint arXiv:1101.0457*, 2011.
- [39] J. Ren, X. Jiang, and J. Yuan, "A fast and accurate cascade subspace face/eye detector on mobile devices," in *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pp. 84–91, IEEE, 2011.
- [40] G. Yang and J. N. Luo, "A real-time face recognition system for android smart phone," *Advanced Materials Research*, vol. 756, pp. 4006–4010, 2013.
- [41] E. Vazquez-Fernandez, H. Garcia-Pardo, D. Gonzalez-Jimenez, and L. Perez-Freire, "Built-in face recognition for smart photo sharing in mobile devices," in *Multimedia and Expo (ICME), 2011 IEEE International Conference on*, pp. 1–4, IEEE, 2011.
- [42] B. Chen, J. Shen, and H. Sun, "A fast face recognition system on mobile phone," in *Systems and Informatics (ICSAI), 2012 International Conference on*, pp. 1783–1786, IEEE, 2012.
- [43] J. Oh, S.-I. Choi, C. Kim, J. Cho, and C.-H. Choi, "Selective generation of gabor features for fast face recognition on mobile devices," *Pattern Recognition Letters*, vol. 34, no. 13, pp. 1540–1547, 2013.
- [44] D. Bong, L. Ngui, and A. Joseph, "Skin color segmentation in mobile platform," in *Industrial Electronics & Applications (ISIEA), 2010 IEEE Symposium on*, pp. 646–650, IEEE, 2010.
- [45] C. Doukas and I. Maglogiannis, "A fast mobile face recognition system for android os based on eigenfaces decomposition," in *Artificial Intelligence Applications and Innovations*, pp. 295–302, Springer, 2010.
- [46] R. F. Sari and P. Indrawan, "Cloud computing services in mobile devices using android face detector api and rest communication," in *The Third International Conference on Digital Information Processing and Communications (ICDIPC2013)*, pp. 529–537, The Society of Digital Information and Wireless Communication, 2013.
- [47] T. Lee and S. Soatto, "Learning and matching multiscale template descriptors for real-time detection, localization and tracking," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pp. 1457–1464, IEEE, 2011.
- [48] X. Yang and K.-T. T. Cheng, "Accelerating surf detector on mobile devices," in *Proceedings of the 20th ACM international conference on Multimedia*, pp. 569–578, ACM, 2012.
- [49] R. B. Gomes, R. V. Aroca, B. M. de Carvalho, and L. M. G. Gonçalves, "Real time interactive image segmentation using user indicated real-world seeds," in *Graphics, Patterns and Images (SIBGRAPI), 2012 25th SIBGRAPI Conference on*, pp. 182–189, IEEE, 2012.
- [50] D. Liu, K. Pulli, L. G. Shapiro, and Y. Xiong, "Fast interactive image segmentation by discriminative clustering," in *Proceedings of the 2010 ACM multimedia workshop on Mobile cloud media computing*, pp. 47–52, ACM, 2010.
- [51] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, vol. 1, pp. 1–511, IEEE, 2001.
- [52] M. Turk and A. Pentland, "Eigenfaces for recognition," *Journal of cognitive neuroscience*, vol. 3, no. 1, pp. 71–86, 1991.
- [53] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 24, no. 7, pp. 971–987, 2002.
- [54] C. Liu and H. Wechsler, "Gabor feature based classification using the enhanced fisher linear discriminant model for face recognition," *Image processing, IEEE Transactions on*, vol. 11, no. 4, pp. 467–476, 2002.
- [55] N. A. bin Abdul Rahman, K. C. Wei, and J. See, "Rgb-h-cbcr skin colour model for human face detection," *Faculty of Information Technology, Multimedia University*, 2007.
- [56] J. Roy and A. Ramanujan, "Understanding web services," *IT professional*, vol. 3, no. 6, pp. 69–73, 2001.
- [57] R. Buyya, J. Broberg, and A. M. Goscinski, *Cloud computing: Principles and paradigms*, vol. 87. John Wiley & Sons, 2010.
- [58] R. Fielding, "Representational state transfer," *Architectural Styles and the Design of Network-based Software Architecture*, pp. 76–85, 2000.
- [59] B. D. Lucas, T. Kanade, et al., "An iterative image registration technique with an application to stereo vision," in *IJCAI*, vol. 81, pp. 674–679, 1981.
- [60] B. M. Carvalho, G. T. Herman, and T. Y. Kong, "Simultaneous fuzzy segmentation of multiple objects," *Discrete Applied Mathematics*, vol. 151, no. 1, pp. 55–77, 2005.
- [61] R. Zabih and V. Kolmogorov, "Spatially coherent clustering using graph cuts," in *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, vol. 2, pp. II–437, IEEE, 2004.
- [62] L. Vincent and P. Soille, "Watersheds in digital spaces: an efficient algorithm based on immersion simulations," *IEEE transactions on pattern analysis and machine intelligence*, vol. 13, no. 6, pp. 583–598, 1991.
- [63] J. Ning, L. Zhang, D. Zhang, and C. Wu, "Interactive image segmentation by maximal similarity based region merging," *Pattern Recognition*, vol. 43, no. 2, pp. 445–456, 2010.
- [64] S. Battiato, G. Farinella, E. Messina, G. Puglisi, D. Rav, A. Capra, and V. Tomaselli, "On the performances of computer vision algorithms on mobile platforms," in *IS&T/SPIE Electronic Imaging*, pp. 82990L–82990L, International Society for Optics and Photonics, 2012.
- [65] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *Computer Vision–ECCV 2006*, pp. 430–443, Springer, 2006.
- [66] M. Agrawal, K. Konolige, and M. R. Blas, "Censure: Center surround extremas for realtime feature detection and matching," in *Computer Vision–ECCV 2008*, pp. 102–115, Springer, 2008.
- [67] S. Srinivasan, Z. Fang, R. Iyer, S. Zhang, M. Espig, D. Newell, D. Cermak, Y. Wu, I. Kozintsev, and H. Haussecker, "Performance characterization and optimization of mobile augmented reality on handheld platforms," in *Workload Characterization, 2009. IISWC 2009. IEEE International Symposium on*, pp. 128–137, IEEE, 2009.
- [68] S. E. Lee, Y. Zhang, Z. Fang, S. Srinivasan, R. Iyer, and D. Newell, "Accelerating mobile augmented reality on a handheld platform," in *Computer Design, 2009. ICCD 2009. IEEE International Conference on*, pp. 419–426, IEEE, 2009.
- [69] M. El Choubassi, O. Nestares, Y. Wu, I. Kozintsev, and H. Haussecker, "An augmented reality tourist guide on your mobile devices," in *Advances in Multimedia Modeling*, pp. 588–602, Springer, 2010.
- [70] Y. Kawano and K. Yanai, "Real-time mobile food recognition system," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2013 IEEE Conference on*, pp. 1–7, IEEE, 2013.
- [71] Y. Kawano and K. Yanai, "Foodcam: A real-time mobile food recognition system employing fisher vector," in *MultiMedia Modeling*, pp. 369–373, Springer, 2014.

- [72] Y. Kawano and K. Yanai, "Foodcam: A real-time food recognition system on a smartphone," *Multimedia Tools and Applications*, pp. 1–25, 2014.
- [73] S. Srinivasan, L. Zhao, L. Sun, Z. Fang, P. Li, T. Wang, R. Iyer, R. Illikkal, and D. Liu, "Performance characterization and acceleration of optical character recognition on handheld platforms," in *Workload Characterization (IISWC), 2010 IEEE International Symposium on*, pp. 1–10, IEEE, 2010.
- [74] A. Vedaldi and A. Zisserman, "Efficient additive kernels via explicit feature maps," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34, no. 3, pp. 480–492, 2012.
- [75] C. Rother, V. Kolmogorov, and A. Blake, "Grabcut: Interactive foreground extraction using iterated graph cuts," in *ACM Transactions on Graphics (TOG)*, vol. 23, pp. 309–314, ACM, 2004.
- [76] F. Perronnin, J. Sánchez, and Y. Liu, "Large-scale image categorization with explicit data embedding," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pp. 2297–2304, IEEE, 2010.
- [77] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1, pp. 886–893, IEEE, 2005.
- [78] A. Ensor and S. Hall, "Gpu-based image analysis on mobile devices," *arXiv preprint arXiv:1112.3110*, 2011.
- [79] C.-H. Chou, P. Liu, T. Wu, Y. Chien, and Y. Zhao, "Implementation of parallel computing fast algorithm on mobile gpu," in *Unifying Electrical Engineering and Electronics Engineering*, pp. 1275–1281, Springer, 2014.
- [80] R. Hofmann, H. Seichter, and G. Reitmayr, "A gpgpu accelerated descriptor for mobile devices," in *ISMAR*, pp. 289–290, 2012.
- [81] N. Singhal, J. W. Yoo, H. Y. Choi, and I. K. Park, "Implementation and optimization of image processing algorithms on embedded gpu," *IEICE TRANSACTIONS on Information and Systems*, vol. 95, no. 5, pp. 1475–1484, 2012.
- [82] N. Singhal, I. K. Park, and S. Cho, "Implementation and optimization of image processing algorithms on handheld gpu," in *Image Processing (ICIP), 2010 17th IEEE International Conference on*, pp. 4481–4484, IEEE, 2010.
- [83] N. Singhal, J. W. Yoo, H. Y. Choi, and I. K. Park, "Design and optimization of image processing algorithms on mobile gpu," in *ACM SIGGRAPH 2011 Posters*, p. 21, ACM, 2011.
- [84] B. Rister, G. Wang, M. Wu, and J. R. Cavallaro, "A fast and efficient sift detector using the mobile gpu," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pp. 2674–2678, IEEE, 2013.
- [85] M. B. López, H. Nykänen, J. Hannuksela, O. Silvén, and M. Vehviläinen, "Accelerating image recognition on mobile devices using gpgpu," in *IS&T/SPIE Electronic Imaging*, pp. 78720R–78720R, International Society for Optics and Photonics, 2011.
- [86] K.-T. Cheng and Y.-C. Wang, "Using mobile gpu for general-purpose computing—a case study of face recognition on smartphones," in *VLSI Design, Automation and Test (VLSI-DAT), 2011 International Symposium on*, pp. 1–4, IEEE, 2011.
- [87] J. Canny, "A computational approach to edge detection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, no. 6, pp. 679–698, 1986.
- [88] T. Strothotte and S. Schlechtweg, *Non-photorealistic computer graphics: modeling, rendering, and animation*. Elsevier, 2002.
- [89] Q. Yang, L. Wang, R. Yang, S. Wang, M. Liao, and D. Nister, "Real-time global stereo matching using hierarchical belief propagation," in *BMVC*, vol. 6, pp. 989–998, 2006.
- [90] C. Harris and M. Stephens, "A combined corner and edge detector," in *Alvey vision conference*, vol. 15, p. 50, Manchester, UK, 1988.
- [91] G.-R. Kayombya, *SIFT feature extraction on a Smartphone GPU using OpenGL ES2. 0*. PhD thesis, Massachusetts Institute of Technology, 2010.
- [92] J. Leskela, J. Nikula, and M. Salmela, "Opencl embedded profile prototype in mobile device," in *Signal Processing Systems, 2009. SiPS 2009. IEEE Workshop on*, pp. 279–284, IEEE, 2009.
- [93] G. Wang, Y. Xiong, J. Yun, and J. R. Cavallaro, "Accelerating computer vision algorithms using opencl framework on the mobile gpu—a case study," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pp. 2629–2633, IEEE, 2013.
- [94] G. Wang, Y. Xiong, J. Yun, and J. R. Cavallaro, "Computer vision accelerators for mobile systems based on opencl gpgpu co-processing," *Journal of Signal Processing Systems*, pp. 1–17, 2014.
- [95] S. H. Kang, S. J. Lee, and I. K. Park, "Parallelization and optimization of feature detection algorithms on embedded gpu," in *International Workshop on Advanced Image Technology*, pp. 164–167, 2014.